



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/631,062	07/31/2003	Bradley Ryan Harrington	AUS920030467US1	3513

35525 7590 02/06/2007
IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER

PANTOLIANO JR, RICHARD

ART UNIT	PAPER NUMBER
----------	--------------

2194

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	02/06/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.		Applicant(s)	
	10/631,062		HARRINGTON ET AL.	
	Examiner		Art Unit	
	Richard Pantoliano Jr		2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 15 November 2006.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 31 July 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.


WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Response to Amendment

1. This Office Action is filed in response to amendments received on **15 November 2006** filed for Application# **10/631,062**. Currently amended **Claims 1, 2, 4, 7, 9, 10, 12, 15, 17, 18, 20, 23** and originally presented **Claims 3, 5, 6, 8, 11, 13, 14, 16, 19, 21, 22, and 24** are currently pending and have been considered below.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. **Claim 1, 6, 9, 14, 17, and 22** are rejected under 35 U.S.C. 103(a) as being unpatentable over Intel ("*Itanium™ Processor Floating-point Software Assistance and Floating-point Exception Handling.*" *Intel Corp., January 2000*) in view of Traut (US Pat: 7,095,705).

4. **Claims 1, 9, and 17:** Intel discloses the method, system and computer readable medium for handling exception vectors by firmware comprising:

a) means and instructions for identifying an exception (*Pg 1-2, ¶4-6, pg 2-2, ¶4-7 and Pg 1-3, Figure 1-1*)(*The emulation library, a software driver means of*

Art Unit: 2194

accessing the firmware, recognizes the exception and passes execution to the firmware, thereby meeting this claim limitation);

b) means and instructions for saving the identified exception (*Page 1-2, ¶4-6*);

c) means and instructions replacing the exception with a substitute exception (*Page 2-2, ¶5 and ¶6*)(*The emulation library can call the kernel exception handler or a user level floating-point exception handler to continue processing the exception*); and

d) means and instructions for restoring the saved exception when control is returned to the operating system (*Page 2-2, ¶4-7*)(*Control is returned to the operating system kernel exception handler before execution of the running program is resumed*).

5. Intel only discloses replacing a single exception and does not disclose the replacement of an entire exception vector. Traut discloses the replacement of an entire exception vector (Col. 6, line 44 – Col. 7, line 45).

6. It would have been obvious to one of ordinary skill in the art at the time of invention to modify the teachings of Intel with the teachings of Traut. One would have been motivated by the desire to allow not only multiple processors in a system on a system access to the updated exception vector, but multiple operating systems running in that system to access those exception vectors, as well (Col. 6, line 44 – Col. 7, line 45).

Art Unit: 2194

7. **Claims 6, 14 and 22:** Intel discloses the method, system and computer readable medium of **Claims 1,9, and 17**, respectively, for handling exception vectors by firmware wherein the data processing system is a symmetric multiprocessor system (*Page 7-2, ¶4*).

8. **Claim 2, 4, 5, 7, 8, 10, 12, 13, 15, 16, 18, 20, 21, 23 and 24** are rejected under 35 U.S.C. 103(a) as being unpatentable over Intel in view of Trout and in further view of Jones (*Jones, Steve. "Using spinlocks in a symmetric multiprocessing environment." Tech Specialist, v2, n10, pg 15(6), Oct 1991*).

9. **Claims 2, 10, and 18:** Intel and Trout disclose the method, system and computer readable medium of **Claims 1,9, and 17**, respectively, for handling exception vectors, but do not disclose the use of a slave loop to suspend the operation of other processors in the system that attempt to access the same portion of firmware until the first process using that code has completed. Jones discloses the use of a spinlock (a slave loop) to pause the execution of a BIOS routine (a form of firmware) by processes other than the one currently processing the BIOS routine (*Jones, Pg 2, last paragraph and Pg 3, 3rd paragraph*). It would have been obvious for one of ordinary skill in the art to combine the teachings of Jones in using spinlocks to pause the execution of other processors in a Symmetric Multiprocessor system while accessing firmware into the system, method and computer readable medium of Intel in order to prevent disruption of computations

Art Unit: 2194

caused by accessing code that should only be accessed by one program at a time (Jones, Pg 1, last paragraph).

10. **Claims 4, 5, 12, 13, 20 and 21:** Intel, Traut and Jones disclose the method, system and computer readable medium of **Claims 2, 10, and 18**, respectively, but Intel does not disclose the storing of the processor context of the process placed in the slave loop or the restoring of said stored state when the processor is able to continue processing. Jones discloses the use of a spinlock (a slave loop) to pause the execution of a BIOS routine (a form of firmware) by processes other than the one currently processing the BIOS routine (Jones, Pg 2, last paragraph and Pg 3, 3rd paragraph). It would have been obvious for one of ordinary skill in the art to combine the teachings of Jones in using spinlocks to pause the execution of other processors in a Symmetric Multiprocessor system while accessing firmware into the system, method and computer readable medium of Intel in order to prevent disruption of computations caused by accessing code that should only be accessed by one program at a time (Jones, Pg 1, last paragraph). The use of a spinlock to pause the execution of a processor inherently requires that the state of the processor be saved because, while the processor keeps checking the state of the lock, it must use registers (memory locations local to the processor that are used to perform operations) previously occupied by the previous task being executed. The contents of the registers that corresponded to the information relating to the exception must be stored in RAM or some other storage medium to allow the processor to continue with that exception, once the lock is held by that processor.

Art Unit: 2194

Once that lock is held, the processor must inherently load that saved context back into the registers in order to continue its previous operation.

11. **Claims 7, 8, 15, 16, 23 and 24:** Intel discloses the method, system and computer readable medium for handling exception vectors comprising:

a) means and instructions for receiving control from an operating system (*Page 1-2, ¶4-6 and Page 2-2, ¶4-7*)(*The state of the interrupted process is saved, and the operating system exception handler passes calls the emulation library to handle the exception*);

b) means and instructions for replacing an exception with substitute code (*Page 2-2, ¶4-7*)(*The state of the interrupted process is saved, and the operating system exception handler passes calls the emulation library to handle the exception*); and

c) means and instructions for restoring the exception when control is returned to the operating system (*Page 2-2, ¶4-7*)(*Control is returned to the operating system kernel exception handler before execution of the running program is resumed*).

12. Intel only discloses replacing a single exception and does not disclose the replacement of an entire exception vector. Traut discloses the replacement of an entire exception vector (Col. 6, line 44 – Col. 7, line 45).

13. It would have been obvious to one of ordinary skill in the art at the time of invention to modify the teachings of Intel with the teachings of Traut. One would have

Art Unit: 2194

been motivated by the desire to allow not only multiple processors in a system on a system access to the updated exception vector, but multiple operating systems running in that system to access those exception vectors, as well (Col. 6, line 44 – Col. 7, line 45).

14. Intel does not disclose means and instructions for suspending execution of any other processor that tries to execute the same code with the use of a slave loop. Jones discloses the use of a spinlock (a slave loop) to pause the execution of a BIOS routine (a form of firmware) by processes other than the one currently processing the BIOS routine (Jones, Pg 2, last paragraph and Pg 3, 3rd paragraph). It would have been obvious for one of ordinary skill in the art to modify the teachings of Jones in using spinlocks to pause the execution of other processors in a Symmetric Multiprocessor system while accessing firmware into the system, method and computer readable medium of Intel in order to prevent disruption of computations caused by accessing code that should only be accessed by one program at a time (Jones, Pg 1, last paragraph).

15. **Claims 3, 11, and 19 are** rejected under 35 U.S.C. 103(a) as being unpatentable over Intel in view of Traut and in further view Jones and in further view of IBM (*IBM Technical Bulletin: NNRD447149. "Method to Prevent Multiple Processes After Taking Exceptions To Enter Open Firmware In a Symmetrical Multiprocessor machine", Published 01 July 2001*).

Art Unit: 2194

Claims 3, 11, and 19: Intel, Traut and Jones discloses the method, system and computer readable medium of **Claims 2, 10, and 18**, respectively, but does not disclose the use of processor identification numbers to identify the processor currently making exclusive use of the firmware and the processors trying to access that same portion of firmware that is already in use. IBM discloses the use of processor IDs in marking the lock used to identify which processor currently holds the lock on the firmware (IBM, pg 1, lines 23-27). It would have been obvious to one of ordinary skill in the art to combine the teachings of IBM into the method, system and computer readable medium of Intel and Jones to ensure that all processors in the system are aware of the fact that the lock is held by a processor and, in particular, which of the other processors in the system is the one that holds that lock.

Response to Arguments

16. Applicant's arguments, see page 7 of Applicant's Remarks/Arguments, filed **15 November 2006**, with respect to the objections to the Specification, the 35 U.S.C. 101 rejections of **Claims 17 and 23**, and the 35 U.S.C. 112, first paragraph rejections of **Claims 4, 12 and 20**, have been fully considered and are persuasive. The objections to the Specification, the 35 U.S.C. 101 rejections of **Claims 17 and 23** and the 35 U.S.C. 112, first paragraph rejections of **Claims 4, 12 and 20** have been withdrawn.

17. Applicant's arguments filed 15 November 2006 have been fully considered but they are not persuasive in regard to the prior art rejections of **Claims 1-24**.

Art Unit: 2194

18. As per **Claim 1**, Applicant argues that Intel does not show passing control “from an operating system to the firmware”. As clarified in the above prior art rejection, the emulation library, a component of the firmware system, in the cited paragraphs are called by the operating system, thereby meeting this claim limitation.

19. Applicant further argues that execution of the emulation library does not constitute control being passed to the firmware. While it is correct that the operating system must make a call to the emulation library in order for an operation to occur, when the library function offered by the emulation library is called, control of execution passes from the operating system kernel to the firmware functions that the library accesses (pg. 7-11, para. 1-2). The purpose of a driver in a computer system is to allow communication and control of the system to be passed between the operating system and the device to which the driver is connected.

20. Applicant further argues that Intel does not show the substitution and restoration of an exception vector. This argument is moot in view of the new grounds of rejection.

21. For the above reasons, Examiner once again rejects **Claim 1**. Since the rejections of **Claims 6, 9, 14, 17, and 22** were argued for similar reasons to **Claim 1**, Examiner rejects **Claims 6, 9, 14, 17, and 22** for the same reasons as **Claim 1** above.

22. As per **Claim 2**, Applicant argues that Examiner, in combining the teachings of Intel with the teachings of Jones, merely shows the protection of a generic data structure during execution and not “placing the processor in a slave loop *until the save exception vector is restored*” or that the “slave loop placement is done in response [to] a

Art Unit: 2194

process *generating an error for an exception vector*". However, the exception vector disclosed by Applicant *is* a generic data structure. Resources such as data structures and memory locations in which data or program code are stored, shared between concurrently operating processors must provide protection around that shared resource when said resource is capable of being accessed by multiple devices at the same time, since modifications to those resources by one processor could adversely affect the operation of the other processors. For this reason the combination of the references is justified. Since, as was stated in the prior art rejection, the exception vector is being modified, it must be protected by some sort of locking mechanism such as a spin lock to ensure that no other device attempted to use the exception vector while it was being replaced. This protection would no longer be needed once the exception vector was replaced. Further, since the passages cited for the rejection of **Claim 1** clearly show the substitution and restoration of exception vectors being performed in response to an exception being generated, all of the limitations of **Claim 2**. Examiner once again rejects **Claim 2** for the above cited grounds.

23. As per **Claims 4 and 5**, Applicant argues these claims on the same grounds as **Claim 2** above. Examiner's reasoning supplied above for **Claim 2** further apply to **Claims 4 and 5**. Applicant further argues that Examiner's contention that the spinlock would *inherently* require a context switch that would require state data to be saved is incorrect. Applicant is correct in stating that a spinlock is a common data structure used to protect data while using minimal processing time in a computer system. However,

Art Unit: 2194

since *any* executing process requires the current state of that executing process to be loaded into the registers of the processor executing that process must be loaded with the state information of that process, no matter how intensive that process is. When replacing one process with another in an executing processor, the state of previous processes must be saved in order to allow continued execution of that preempted process at a later time. Therefore, Examiners contention of inherency in the storage of state data in regard to **Claims 4 and 5** is valid and Examiner rejects **Claims 4 and 5** for the above reasoning.

24. As per **Claims 10, 12, 13, 18, 20, and 21**, the rejection of these claims were argued for the same reasons as **Claims 2, 4 and 5** above. Therefore, **Claims 10, 12, 13, 18, 20, and 21** are also rejected for the above stated reasoning as **Claims 2, 4, and 5**.

25. As per **Claim 7**, Applicant argues that Intel fails to teach restoring the exception vector when control is returned. This argument is moot in view of the new grounds of rejection. Further, Applicant argues that there is no showing of suspending processors when encountering substituted code. Resources such as data structures and memory locations in which data or program code are stored, shared between concurrently operating processors must provide protection around that shared resource when said resource is capable of being accessed by multiple devices at the same time, since modifications to those resources by one processor could adversely affect the operation

Art Unit: 2194

of the other processors. For this reason the combination of the references is justified. Since the claimed processor is accessing a shared resource, the use of a spinlock to protect that shared resource from other processors while the claimed processor is executing said code is a valid combination. Examiner therefore finds Applicant's argument in regard to **Claim 7** unpersuasive and rejects **Claim 7** based on the above reasoning.

26. As per **Claims 3, 11, and 19**, the rejection of these claims was argued using the same reasoning as for the rejection of **Claim 2**. Therefore, Examiner rejects **Claims 3, 11, and 19** for the reasons cited above for **Claim 2**.

Conclusion

27. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure: Konopik et al (US Pat: 4,768,149), Jons et al (US Pat: 5,065,354), Liu et al (US Pat: 5,446,877), Brewer et al (US Pat: 5,455,919), Fisherman et al (US Pat: 5,586,301), Favor (US Pat: 5,794,063), Novak et al (US Pat: 5,909,567), Dawson (US Pat: 6,397,382), and Bourekas (US Pat: 7,133,951).

28. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Richard Pantoliano Jr whose telephone number is (571) 270-1049. The examiner can normally be reached on Monday-Thursday, 8am - 4 pm EST.

Art Unit: 2194


29. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on (571)272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

30. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

RP

RP

02/05/07


WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER